REMARKS

Claims 1-16 are pending in the present application. Claims 1 and 7 are the independent claims. In the Official Action, dated December 9, 2003, claims 1-16 were rejected under 35 U.S.C § 103(a) as allegedly obvious over U.S. Patent No. 6,185,613 B1 ("Lawson") in view of U.S. Patent No. 5,603,034 ("Swanson").

The specification has been amended herein to correct matters of minor informality, e.g., spelling, grammar, idiom and the like. No new matter was added. To clarify the invention, independent claims 1 and 7 have been amended herein. Dependent claims 2-6 and 8-16 have also been variously amended herein, where required, to properly refer to the independent claims.

Without conceding the propriety of the combination being applied in forming the outstanding rejection, in light of the amendments and the accompanying remarks below, Applicants respectfully submit that the claims now patentably define over the prior art and are in condition for allowance.

The following format will be used to cite references in this response: The application for the present invention will be referred to as "App." followed by the appropriate page and line number, e.g., "App. 2:26-2:28" refers to lines 26-28 on page 2 of the Application. The outstanding Official Action will be referred to as "Official Action" followed by the page and item number, e.g., "Official Action 2:3" refers to item 3 on page 2 of the Official Action. Patents will be referred to by an inventor, as set forth above, followed by a column and line number, e.g., "Lawson 3:27" refers to Lawson, column 3, line 27.

Summary of the Invention

Distributed computing systems need to ensure that they can function even when portions of the system fail or go off-line. App. 2:13-2:14. For example, when one object goes down because of a failure of one computer, the objects that reference that failed object should not also fail. App. 2:14-2:16. In addition, when the failed object comes up, the objects that reference that failed object should be able to continue to access the object. App 2:16-2:17. For the purpose of referring to the claims, note that the terms "object" and "resource" are used interchangeably. App. 14:27-15:1.

The solution provided by the present invention includes "nodes" and a bus manager that are interconnected through a communication link or bus. App. 15:2-15:3. Each node may contain resources that request access to other resources. A resource that requests access to another resource is referred to as a client resource or client, and a resource that is accessed by another resource is referred to a server resource or server. The bus manager tracks all the resources as they come up and go down. Each node on which a resource is located, referred to as a server node, notifies the bus manager when a resource comes up and when a resource goes down. App. 15:4-15:9.

The watching of a resource is coordinated by the bus manager, but the monitoring of a resource is performed on a client node to server node basis without interaction from the bus manager. When a client wants to watch a resource so that it knows when the resource is in the up state, the client node notifies the bus manager. The resource is identified using a tracking reference. If the resource is already up, then the bus manager notifies the client node

that the resource is up. Otherwise, the bus manager notifies the client node when the resource comes up. When the client node is notified that the resource is up, it may notify the bus manager to stop watching for the resource. **The monitoring of a resource is performed on a peer to peer basis. That is, once a client node is informed that a resource has entered the up state, it establishes a connection directly with the server node that contains the resource. Once the connection is established, the client node notifies the server node periodically that it is still up and running. If the server node does not receive this notification, it assumes that the client node is no longer up and running and resets its internal state accordingly.** Similarly, if the client node receives an error when sending its periodic notification to the server node, it assumes the server node is down and resets its internal state accordingly. When the resource goes down, the server node notifies the client node that the resource is now down. App. 15:13-16:2.

Root Reference Lawson

Lawson is directed to a similar problem as that of the present invention, however, the motives and corresponding solutions of Lawson are quite different. A primary goal of Lawson was to provide systems and methods for global event notification that allow a local event consumer to receive notification of events without registering each individual server in the network. Lawson 6:28-6:32. Lawson was also motivated to minimize the amount of message traffic generated on the network. Lawson 2:34-2:35. As such, Lawson provides a system and method for globalizing event notifications in a distributed computing environment. Lawson 4:35-4:36.

Lawson achieves global event notification by storing a global event registry comprising a list of events and a corresponding list of servers which need notification when the corresponding event occurs. Lawson 4:54-4:57. In addition to the global registry, each server stores a local event registry comprising a list of events and a corresponding list of local event consumers that need notification when an event occurs. The basic event globalization process utilized [sic] these two registries to globalize events as follows:

Each server has running thereon a local event globalization process that registers for events desired by local event consumers. When an event consumer registers for an event, the event globalization process of the event consumer's local server places an entry into the local event registry for the local server. An entry is also placed into the global event registry for the local server where the event consumer is located. Thus, the global event registry is updated to contain an entry identifying the server where the event consumer is located and the desired event and the local event registry of that server is updated to identify the event consumer and the desired event. The above description can be found at Lawson 4:57-5:7.

Rejection of Claims 1-16 under 35 U.S.C. § 103(a)

As stated above, Claims 1-16 were rejected under 35 U.S.C. § 103(a) as allegedly obvious over Lawson in view of Swanson. Applicants respectfully traverse the outstanding rejection for at least the following reasons.

Rejections under 35 U.S.C. 103(a) require that the reference (or combination of references) disclose every element of the claim. As stated in the MPEP, "the prior art references

(or references when combined) must teach or suggest all the claim limitations." MPEP § 706.02(j). Without conceding that an event as disclosed by Lawson can be considered the same as a notification as taught by the present invention, Applicants have amended the claims herein and hereby present the following remarks.

First, Applicants respectfully submit that neither Lawson nor Swanson teaches or suggests monitoring a state of a second software component, as recited in claim 1.

Claim 1, as amended, recites the following:

> 1. (previously presented) A method in a computer system for providing property notifications for properties of software components in a distributed computing environment, the method comprising:
> registering, by a first software component, an interest in watching a property of a second software component;
> receiving a notification when the property is set;
> ~~tracking~~ **monitoring a state of the second software component, wherein monitoring a state comprises sending periodic notifications to the second software component**; and
> determining when the second software component is in a down state based upon said ~~tracking~~ monitoring.

(emphasis added).

By "monitoring a state of the second software component," the present invention is capable of maintaining a peer to peer connection between a first software component and a second software component. App. 15:20-15:21. As stated in claim 1, this comprises sending periodic notifications to the second software component. App. 15:23-15:24. When a server resource goes down, the client resource is informed, for example, by the failure of the server resource to respond to the periodic notifications. App. 15:24-15:26.

In contrast, Lawson establishes a hierarchical model for event notification. See description above. As explained, in Lawson, local event consumers register for notification with a server, and the server is registered with a global event registry. The computer that houses the global registry notifies appropriate servers on the occurrence of an event, and the servers subsequently notify the appropriate local event consumers. Accordingly, Lawson does not provide for monitoring a state of the second software component, wherein monitoring a state comprises sending periodic notifications to the second software component. Instead, event consumers receive communications about a particular resource only on the occurrence of an event. *See* Lawson 4:42-4:46 ("event notification system...sends notification of events that occur").

Swanson likewise does not provide for monitoring a state of the second software component, wherein monitoring a state comprises sending periodic notifications to the second software component, as recited in claim 1. Swanson is directed to altering graphical interfaces for software applications. A graphical resource editor provides users with the capability to access application resources to change one or more attributes of an application's graphical user interface. *See* Swanson 6:48-6:56. As such, Swanson is not directed to a similar problem as that of Lawson and the present invention. Accordingly, Applicants respectfully submit that application of Swanson to the invention does not cure the above-identified deficiency of Lawson with respect to claim 1. More particularly, Applicants respectfully submit that Swanson does not teach or suggest monitoring a state of the second software component, wherein monitoring a state comprises sending periodic notifications to the second software component.

As claims 2-6 and 9-11, 13, and 15 depend either directly or indirectly from claim 1, they are believed allowable for the same reasons. Withdrawal of the rejection of claims 1-6, 9-11, 13 and 15 under 35 U.S.C. § 103(a) is thus earnestly solicited.

Second, Applicants respectfully submit that claim 7 contains an element similar to the "monitoring a state of the second software component, wherein monitoring a state comprises sending periodic notifications to the second software component" of claim 1, and is therefore allowable on similar grounds. Claim 7, as amended, provides:

> 7. (previously presented) A method in a computer system for providing property notifications for property settings in a distributed computing environment, the method comprising:
>
> for each of a plurality of software components, registering an interest in a property, wherein the property is associated with a server node; ~~and~~
>
> **periodically notifying the server node that a software component is operational, if said software component is operational;**
>
> setting the property a plurality of times; and
>
> for each setting of the property, notifying each software component of the plurality of software components that the property has been set prior to notifying any software component of the plurality of software components of any later setting of the property.

(emphasis added).

Claim 7 recites "periodically notifying the server node that a software component is operational, if said software component is operational." This limitation is not disclosed in any of the references of record, taken alone or in combination.

As described, Lawson sends event notifications to event consumers, but does so on the occurrence of an event and not through periodic communications. In fact, by suggesting that reduction of network traffic is a goal of the system (see above), Lawson discourages the periodic notifications conducted in accordance with the present invention and thus Lawson is believed to

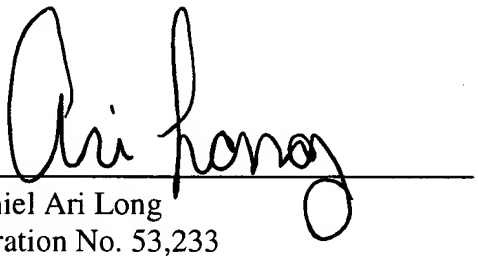
teach away from performing the present invention. As described above, Swanson does not cure Lawson's deficiency with respect to monitoring in the manner performed by the invention.

Accordingly, it is respectfully submitted that independent claim 7 is patentable over Lawson and Swanson, whether taken alone or in combination. As claims 8, 12, and 14 depend either directly or indirectly from claim 13, they are believed allowable for the same reasons. Withdrawal of the rejections under 35 U.S.C. § 103(a) is thus earnestly solicited.

CONCLUSION

Applicants believe that the present reply is responsive to each of the points raised by the Examiner in the Office Action, and submit that Claims 1-16 of the application are in condition for allowance. Favorable consideration and passage to issue of the application at the Examiner's earliest convenience is solicited in earnest.

Date: March 4, 2004



Nathaniel Ari Long
Registration No. 53,233

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439